Journal of Advances in Science and Technology

Vol. III, No. VI, August-2012, ISSN 2230-9659

OBJECT ORIENTED DESIGN PATTERNS AND FRAMEWORKS

www.ignited.in

Journal of Advances in Science and Technology
Vol. III, No. VI, August-2012, ISSN 2230-9659

Object Oriented Design Patterns and Frameworks

Naman[1] Pratyush[2]

[1]Software Engineer HCL Technologies, Noida

[2]B-Tech, Information Technology Asansol Engineering College, Asansol, WB

Abstract: ***-.Object-oriented analysis (OOA) and Object-oriented design (OOD) are the techniques that have been since 1988 to support the object-oriented software development with the help of languages like Smalltalk, Eiffel and C++. Object oriented analysis and design is concerned with the software engineering requirements and specifications that help to programmers to solve complex programs.***

Key Words- Object Oriented Design, Object Oriented Framework, OOA, Object Oriented Analysis, Smalltalk.

----------♦----------

1.1 AN INTRODUCTION TO OBJECT ORIENTED SOFTWARE DEVELOPMENT

The object oriented model was developed for solving complex problems. The object oriented model resulted in the object oriented programming paradigms. In 1980s, object oriented software development was started. Object oriented programming seems to be more effective in solving complex problems in the software industries. Both the software professionals and the end users are benefited by the concept of object oriented programming. The object oriented programming provides excellent means of communication among the designers, analysts, programmers and end users.

The convergence of object-oriented models, techniques, development of object oriented design patterns and frameworks and evolution of object oriented programming languages are the main process that helps for the Object Oriented Software Development (OOSD). Object-oriented model is designed to implement various techniques and framework for the software development. Software is developed by using the life cycle model. Software life cycle process defines how the software product must be developed and maintained (Johnson, 1988). Object oriented approach mainly focuses on improving the reusability and maintainability of the software systems by using the set of techniques, tools, notations and criteria. The following figure illustrates the object oriented software development model:

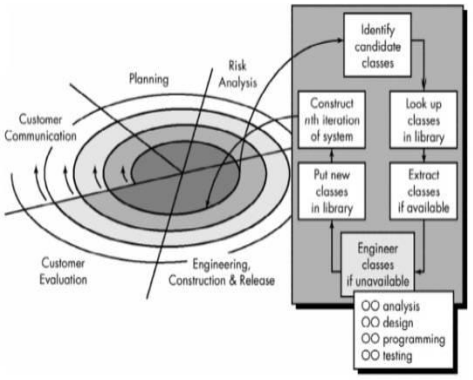

Figure 1: Object oriented software development model

Source: Laganeire and Lethbridge, 2003

The object oriented approach uses various principles, patterns, designs, models and frameworks to develop the software. Particularly, the object oriented approach helps to write well-designed and medium-sized object-oriented programs to develop the software. The software development is the framework that involves set of tasks which are required to develop the software system. Object Oriented Software Development is basically depend on the object oriented life cycle model and it involves various processes such as components, classes, interfaces, attributes, objects, functions and operations which will satisfy the requirements for developing the software.

WHAT IS OBJECT ORIENTED DESIGN

Object-oriented design is concerned with developing the object oriented model of the software system to implement the requirements. The object-oriented design's objects are related to the solution to the problem which is solved. There will be some relationship between some solution objects and some

problem objects but the designer should have ability to add new objects and to change the problematic objects to find the solution. The input for OOD (object oriented design) is provided by the output of OOA (object oriented analysis). Object oriented design is one of the processes of planning a system of the interacting objects to solve the software problem.

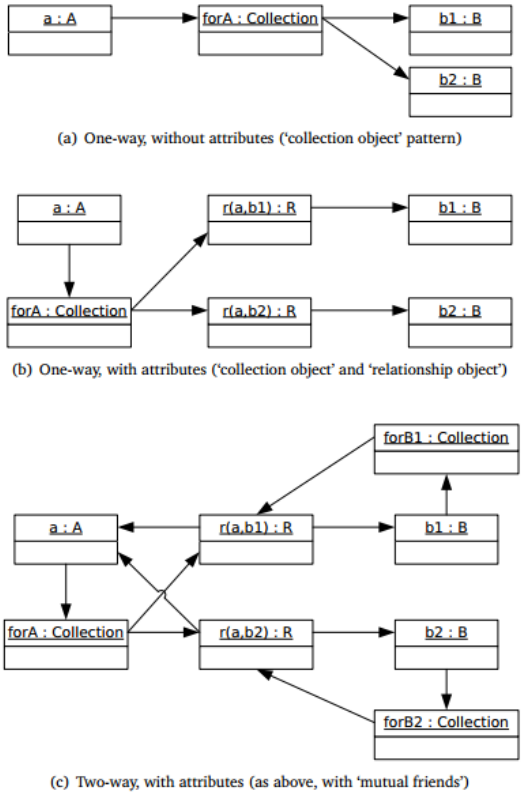

Figure 2: Objects representing relationship

Source: Wren, 2007

Object oriented design is one of the approaches of object oriented software design. Object-oriented design is used to create the architecture for software implementation. Objects in the Object Oriented Design are the abstractions of real-world entities or system and they have capacity to manage themselves. Objects in the Object Oriented Design can communicate by message passing and they are independent. The system functionalities are expressed in terms of the object. Object-oriented design means designing the software in which the fundamental components of the design will represent various objects with their operations and functions.

Object oriented design is represented in terms of classes and objects and the relationships among them. Object Oriented Design (OOD) is the concept that makes the programmers to plan their code correctly to have better program to make the software. Simula and SmallTalk are the first languages that supports to the object oriented design. Grady Booch wrote the paper in 1992 with title "Object-Oriented Design" and that makes the term Object Oriented Design (OOD) to become popular. Some of the tools for Object Oriented Design are Coad-Yourdon; Rambaugh (OMT); Booch and UML. Some of the languages that supports for Object Orientation (OO) are C++; Smalltalk; Eiffel; Object C; Object Pascal and Java.

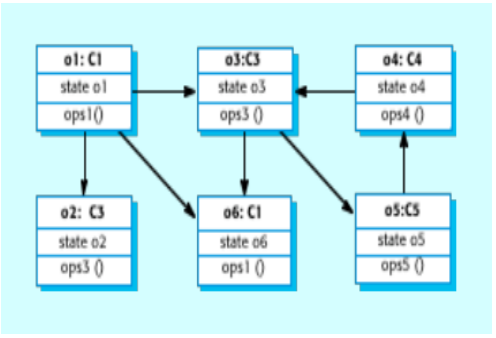

Figure 3: Example for objects in OOD

Source: Sommerville, 2004

Object Oriented Design is defined as the programming language and it has five conceptual tools that helps programmer to develop the software (Johnson and Richard, 1999). Encapsulation, Data protection, Interface, Inheritance and polymorphism are five main concepts of the Object Oriented Design (OOD) and these concepts are mainly used in the implementation levels to build the programming language. These programs are more readable than the non-object oriented programs. Object Oriented Design brings several advantages to the Object Oriented Software Development such as privacy, documentation and re-usability.

An overview of existing models for object oriented analysis and design

Object-oriented analysis and design is the software engineering approach which has ability to model the system as the group of interacting objects (Ambler and Scott, 2000). Each object will represent some entity of the system which is being modeled and characterized by its class and behavior. Object-oriented analysis and design allows creating various models to show the dynamic behavior, static structure and run time deployment of the collaborating objects. OOAD project phases are requirement gathering, analysis, and modeling.

The following are some of the models that are related to the OOAD (object-oriented analysis and design).

- Functional Decomposition model
- Structured Analysis model.

- Data modeling model.
- Object oriented model.

UML (Unified Modeling Language) is one of the object-oriented analysis and design model (Chonoles, Jesse and Schardt 2003). Unified Modeling Language is the standardized general-purpose modeling language and it is the field of object oriented software engineering. It includes class diagram, component diagram, object diagram, profile diagram, package diagram and composite structure diagram.

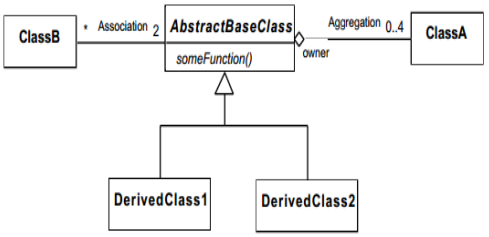

Figure 4: Class diagram

Source: Beall, 1999

WHAT IS THE UNIQUENESS OF OBJECT ORIENTED FRAMEWORK

A framework is the abstract design for a specific application, and it usually contains number of classes and these classes will be taken from the class library or application specific (Fayad, Schmidt, Johnson, 1997). The framework may consist of abstract classes and the operations in which they implement, and also the expectations on the concrete subclasses (Butler, n.d). Object oriented frameworks play an important role in enabling the technology for reusing both the functionality of software components and the architecture. Typically, frameworks have the steep learning curve that the user should understand the abstract design of object collaboration rules and underlying framework.

Object-oriented frameworks are the promising technology for software designs and implementations to improve the quality of software and also to reduce the cost. A framework is the semi-complete and reusable application which can be specialized to produce many custom applications (Fayad, Schmidt, and Johnson, 1997; Johnson, and Foote, 1998). The primary benefits of Object orientation application frameworks are the modularity, extensibility, reusability, and inversion of control.

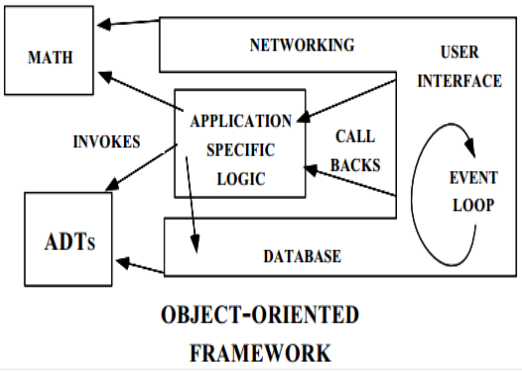

Figure 5: Object Oriented Framework

Source: Schmidt (n.d.)

AN INSIGHT OF VARIOUS DESIGN PATTERNS IN OBJECT ORIENTED FRAMEWORK:

A framework is the integrated collection of components which collaborate to produce the reusable architecture for related applications. Design patterns are used to represent the solutions to problems that arise during the software development within the particular context. Patterns may capture both the dynamic and static structure and also the collaboration among the key participants in the software designs.

Design patterns have ability to enable large-scale reuse of the software architectures. Design Patterns may capture the expert knowledge and design and also it helps to improve developer the communication. Design Patterns help for the ease the transition to the object oriented technology.

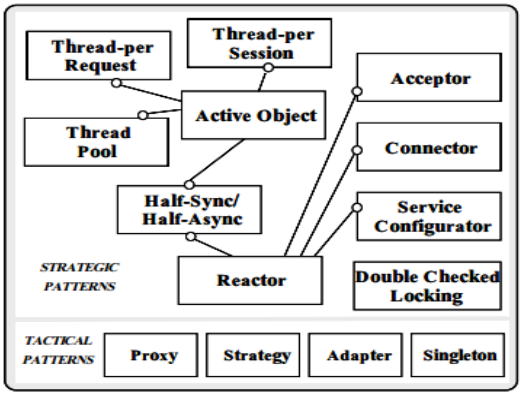

Figure 6: Design Patterns in the Blob Server

Source: Schmidt (nd)

Design patterns are used in the object-oriented frameworks.

There are two kinds of design patterns:

1. Extensible design patterns and

2. Static design patterns

Applications that are developed by using the extensible design patterns are very difficult to test and this is due to the communication complexity, dynamic binding, dynamic typing and extensibility.

- Programming languages and software used in applying object oriented design
- Simula is the first language with primary features of object-oriented language and it was developed in the year 1967.
- Smalltalk is developed with the features of object-oriented language (1972 to 1980)
- Pure object-oriented language has classes, modules, prototypes, blocks, etc and they are designed to facilitate object-oriented methods Examples: Emerald, Eiffel, Scala, Obix, JADE, and Smalltalk
- Languages that are designed mainly for object-oriented programming C++, Java, VB.net, C#, Python
- Languages with some features of objects oriented (methods, classes, reusability, inheritance): Common Lisp, Oberon: Oberon-1 or Oberon-2
- Languages with historically procedural languages but extended with some objects oriented features are VB, COBOL, PHP, FORTRAN, ABAP.

CONCLUSION

Object-oriented programming languages become popular since 1990s. The object oriented software development is the processes that carrying out the development functions and activities for the successful deliverables of the projects. The object oriented software development is fully carried out by the life cycle model which breaks down the development phases into many stages and phases to complete the process of software development. Object oriented design and object oriented analysis is the process that mainly helps to complete the object oriented software development process successfully. The object oriented software development is a boon or the programmers in many ways by providing modularity, extensibility, inversion of control and reusability. Throughout the evolution of object oriented programming language it provides various advantages to the object oriented software development.

1.8 REFERENCES

1. Sommerville I, (2004): Software Engineering, 7th edition, Object-oriented Design, Chapter 14.
2. Ambler, Scott, (2000): Take an Object-Oriented Approach to Analysis, Computing Canada, July 21, 2000.
3. Johnson, Richard, (1999): An Industry Analysis of Developer beliefs about Object-Oriented Systems Development, Database for Advances in Information Systems, Winter, 1999.
4. Chonoles, Jesse M; Schardt J A (2003): UML 2 for Dummies, Wiley Publishing
5. M.E. Fayad, D.C. Schmidt, R.E. Johnson, (1997): “Building Application Frameworks”, Addison-Wesley, 1999. Special Issue of CACM, October 1997
6. Butler G (n.d): Object-Oriented Frameworks, retrieved on 25th October 2012 from http://users.encs.concordia.ca/~gregb/home/PDF/ecoop-tutorial.pdf
7. Beall M W, (1999): An Object-Oriented Framework for the Reliable Automated Solution of Problems in Mathematical Physics, Rensselaer Polytechnic Institute Troy, New York.
8. Johnson, R.E. and Foote, B. (1998): Designing reusable classes. J. Object-Oriented Programming 1, 5 (June/July 1988), 22–35
9. Fayad, M.E., Schmidt, D.C., and Johnson, R.E (1997): Object-Oriented Application Frameworks: Problems and Perspectives. Wiley, NY, 1997, to appear
10. Schmidt (nd): Design Patterns and Frameworks for Object-oriented Communication Systems, Washington University, St. Louis.
11. Johnson (1988): Ralph Johnson and Brian Foote. ``Designing Reusable Classes.'' Journal of Object-Oriented Programming. SIGS, 1, 5 (June/July. 1988), 22-35.
12. Wren A, (2007): Relationships for object-oriented programming languages, Cambridge CB3 0FD, United Kingdom

13. Leithbridge T C and Laganaire R (2001), Object-Oriented Software Engineering:

14. Practical Software Development using UML and Java, McGraw Hill, USA